Report -1

Submitted by Sarbojit Das

May 16, 2024

1 Dataset

The `advertising` data set, contained in the `glmtoolbox` package in `R`, consists of sales of that product in 200 different markets. It also includes advertising budgets for the product in each of those markets for three different media: TV, radio and newspapers.

Following is an extended description of the terminologies associated with this dataset:

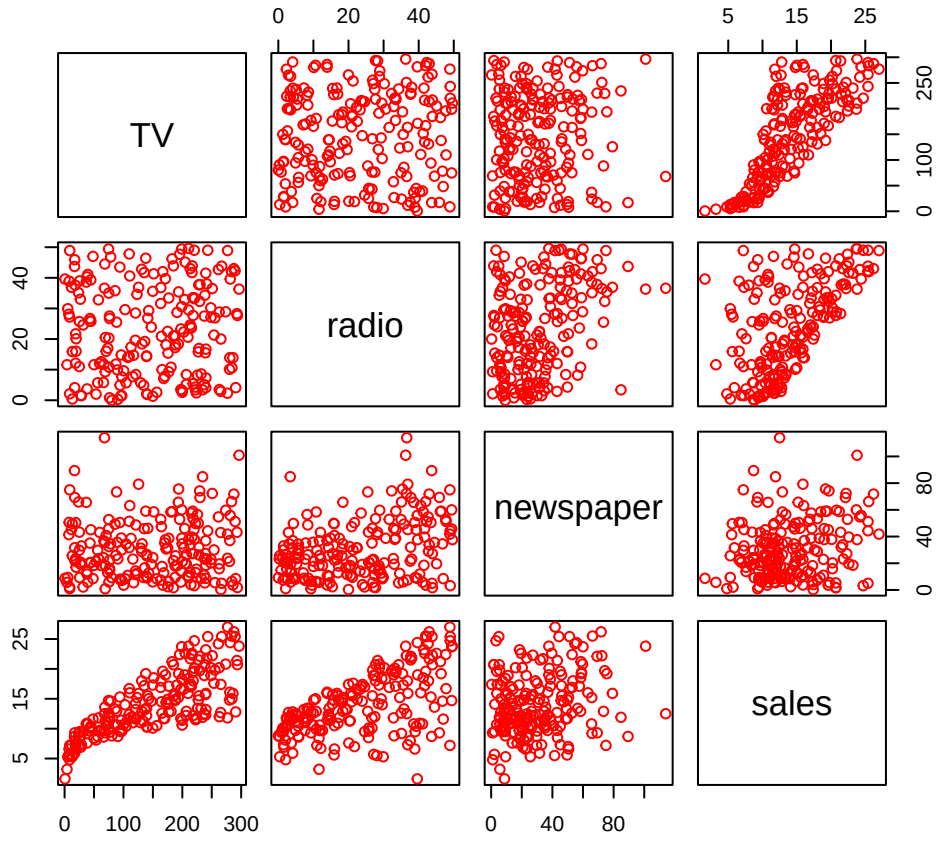
1. TV: a numeric vector indicating the advertising budget on TV.
2. radio: a numeric vector indicating the advertising budget on radio.
3. newspaper: a numeric vector indicating the advertising budget on newspaper.
4. sales: a numeric vector indicating the sales of the interest product.

First of all, we need to load our dataset from the `glmtoolbox` package in `R`.

```
library(glmtoolbox)
data(advertising)
dat <- advertising
```

2 Exploratory Data Analysis (EDA)

We plot the data to understand characteristics about it. Plotting a scatterplot of the features.



From the plot, we see that:

1. There's somewhat of a positive linear trend for TV vs sales and radio vs sales.
2. Clustering of observations towards the center occur for newspaper vs sales. Hence, columns are very weakly related with each other. So. correlation might be closer to zero.

Hence, we are thereby motivated to build a simple linear regression model (SLRM) considering response y as sales and covariates: newspaper, radio and TV respectively.

To justify our claim, we compute the correlation matrix between the columns as follows:

Table 1: Correlation matrix for advertising dataset

	TV	radio	newspaper	sales
TV	1.000	0.055	0.057	0.782
radio	0.055	1.000	0.354	0.576
newspaper	0.057	0.354	1.000	0.228
sales	0.782	0.576	0.228	1.000

3 Simple Linear Regression Model

Consider, the design matrix given by $X = (\underset{\sim}{X_1}, \underset{\sim}{X_2}, \underset{\sim}{X_3})$ denoting the vector of predictors, namely, radio, sales and newspaper respectively for n sampled observations.

Assumptions:

Linearity	The relationship between X_i (covariates) and y (response) must be linear for $i = 1, 2, 3$.
Independence of errors	There is not a relationship between the residuals and the y (response) variable; in other words, is independent of errors.
Normality of errors	The residuals must be approximately normally distributed.
Equal variances	The variance of the residuals is the same for all values of X_i (covariates) for $i = 1, 2, 3$.

We define our SLRM model considering sales as reponse, y and radio as covariate, X_1 with the noise term ϵ. Both the variables are continuous in nature. For n observations, our model is defined as follows:

$$y_j = \beta_0 + \beta_1 X_{1j} + \epsilon_j \text{ for } j = 1, \cdots, n$$

where β_0 and β_1 are the coefficients to be estimated.

In `R`, we use the `lm()` function to fit the simple linear regression model.

```
lm.fit <- lm (sales ~ radio, data = dat)
```

Summary statistics of our model: For this, we use the `summary()` function in `R`.

```
summary(lm.fit)
```

```
Call:
lm(formula = sales ~ radio, data = dat)

Residuals:
     Min       1Q   Median       3Q      Max
-15.7305  -2.1324   0.7707   2.7775   8.1810

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.31164    0.56290  16.542   <2e-16 ***
radio        0.20250    0.02041   9.921   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 4.275 on 198 degrees of freedom
Multiple R-squared:  0.332, Adjusted R-squared:  0.3287
F-statistic: 98.42 on 1 and 198 DF,  p-value: < 2.2e-16
```

Fitted model's coefficients: The obtained estimated coefficients are:

	coef.lm.fit.
(Intercept)	9.3116381
radio	0.2024958

3.1 Visualisations

3.1.1 Plot - 1

```
plot(dat$radio, dat$sales, pch = 20,
     xlab = "radio", ylab = "sales", col = "blue")
points(dat$radio, predict(lm.fit), pch = 20, col = "orange")
abline(lm.fit, col = "red", lty = 2, lwd = 2)
legend ("topleft", legend = c("Observed Data", "Predictions", "Fitted Regression line"),
        col = c("blue", "orange", "red"), pch = c(15,15, 19), lty = c(NA, NA, 2),
        lwd = c(NA, NA, 2), cex = 0.7, pt.cex = c(1.2,1.2,0.2))
```

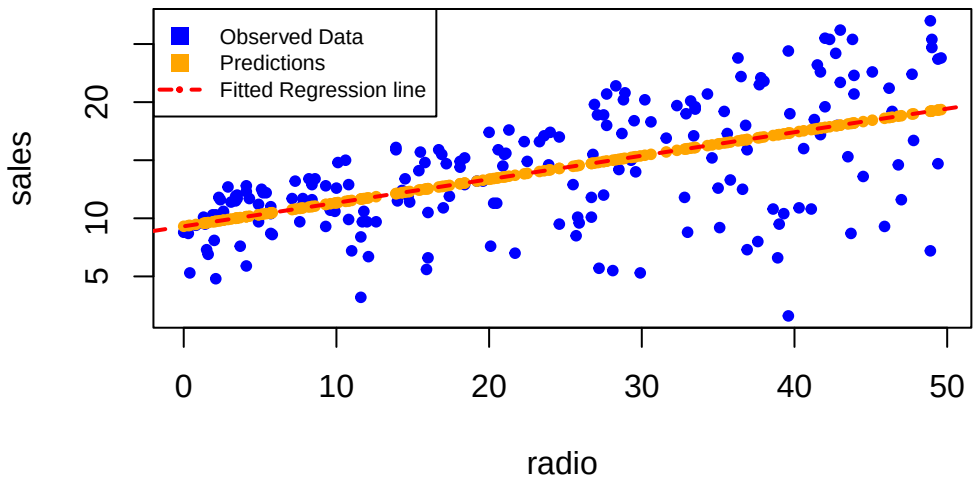


Figure 1: Plot of regression line fitted to data

3.1.2 Plot - 2

```
plot(predict(lm.fit), residuals(lm.fit),
     col = "purple", pch  = "*",
     xlab = "Predicted sales", ylab = "Residuals")
```

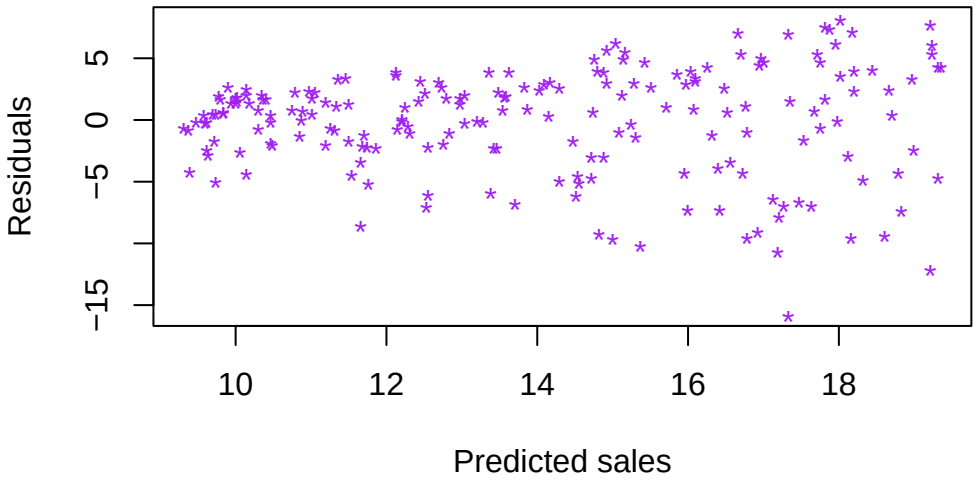


Figure 2: Plot of predicted response (fits) against residuals

The scatter plot does not show any pattern/shape. Hence, variance of the residuals should be same across all values of the X-axis and the correlation should be approximately 0. Thus, the 2^{nd} and 4^{th} assumptions have been been met.

3.1.3 Plot - 3

```
hist(residuals(lm.fit), col = "red",
     border = "white", main = "Histogram of residuals",
     xlab = "Residuals", freq = FALSE)
lines(density(residuals(lm.fit)), col = "blue", lty = 2, lwd = 2)
```

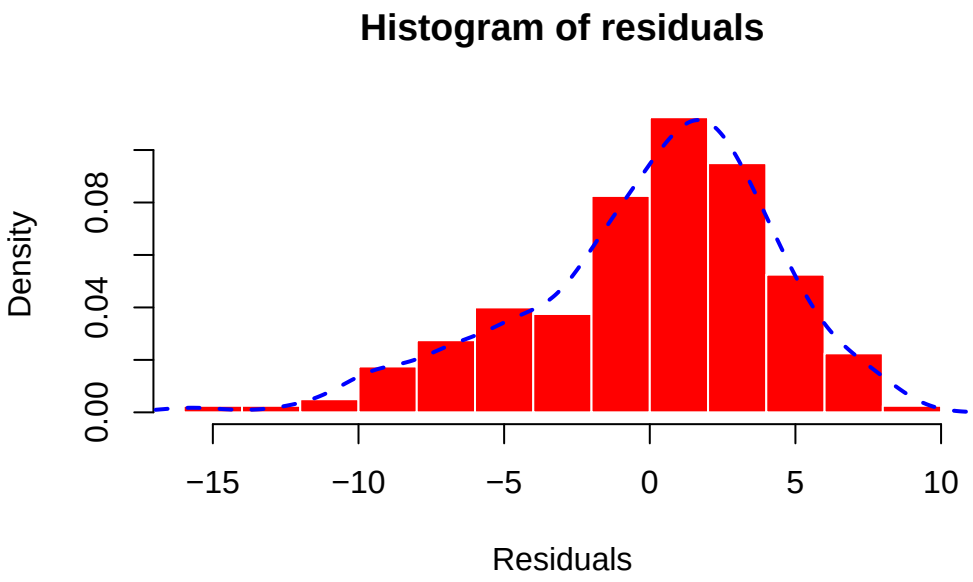


Figure 3: Histogram

From the histogram of the residuals, we conclude that it is approximately normally distributed. Hence, the 3^{rd} assumption is met.

4 Binary Classification:

We fix a small window of values around X_1.

```
index <- dat$radio > 10 & dat$radio <= 35
hist(dat$sales[index], col = "green", main = "Histogram of sales given radio",
     xlab = "Sales conditioned on radio")
```

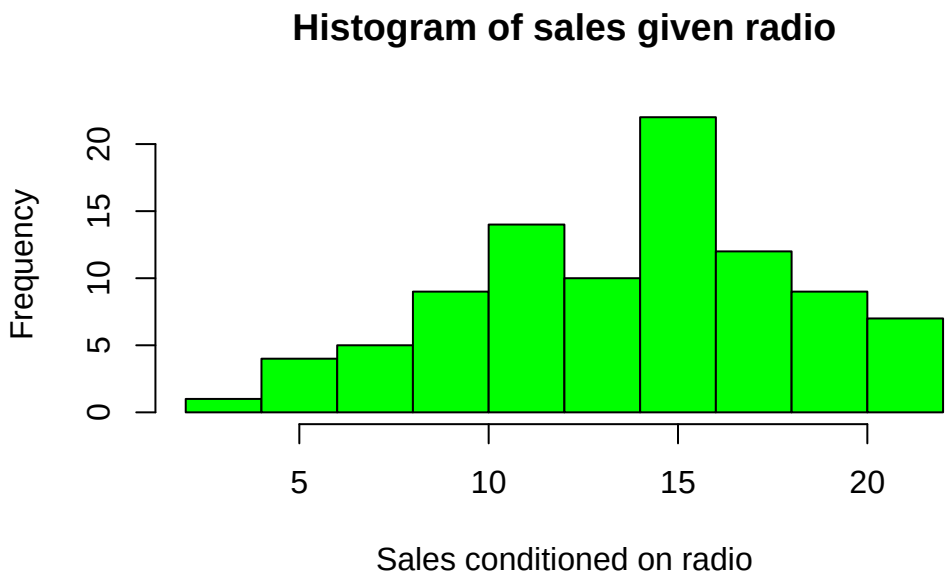


Figure 4: Conditioned Histogram of Sales

From the histogram, we see that the conditioned distribution of sales given radio is approximately normal.

1. Quantiles for our response:

```
   0%    25%    50%    75%   100%
1.600 10.375 12.900 17.400 27.000
```

Converting continuous response to discrete type

Rule: We set a threshold of sales value as 17. If observations in response is greater than the aforementioned threshold, then we assign it to Class 1, else, to Class 0.

```r
y <- ifelse (dat$sales > 17, 1, 0)
```

2. Finding the estimated probabilities of classification:

```r
# Setting up our design matrix
X <- cbind(1, dat$radio)
# Estimated beta
beta.hat <- matrix (solve(t(X) %*% X) %*% t(X) %*% y, ncol = 1)
# Estimated probabilities of classification
pi.hat <- exp(X %*% beta.hat)/(1 + exp(X %*% beta.hat) )
qi.hat <- 1 - pi.hat
```

```r
predict.probs <- data.frame(pi.hat ,qi.hat)
names(predict.probs) <- c("Class 1", "Class 0")
kable(head(predict.probs), caption = "Snapshot of Predicted Probabilities")
```

Table 5: Snapshot of Predicted Probabilities

Class 1	Class 0
0.6300235	0.3699765
0.6360718	0.3639282
0.6621563	0.3378437
0.6440695	0.3559305
0.5159067	0.4840933
0.6737055	0.3262945

3. Computing the predicted response for a particular threshold:

```r
threshold <- 0.60
predict.y <- ifelse(pi.hat > 0.60, 1, 0)
```

4.1 Confusion Matrix

Now, we compute the confusion matrix.

```r
foo <- data.frame(y, predict.y)
tp <- sum(foo[ ,1] == 1 & foo[ ,2] == 1)
fn <- sum(foo[ ,1] == 1 & foo[ ,2] == 0)
fp <- sum(foo[ ,1] == 0 & foo[ ,2] == 1)
tn <- sum(foo[ ,1 ] == 0 & foo[, 2] == 0)
confusion.matrix <- matrix(c(tp, fn, fp, tn), nrow = 2, ncol = 2, byrow=TRUE)
colnames(confusion.matrix) <- c("Predicted: Class 1", "Predicted: Class 0")
rownames(confusion.matrix) <- c("Actual: Class 1", "Actual: Class 0")
confusion.matrix
```

```
               Predicted: Class 1 Predicted: Class 0
Actual: Class 1                 41                 15
Actual: Class 0                 28                116
```

4.2 Sensitivity and Specificity

```
tpr <- tp/(tp + fn)
fpr <- fp/(fp + tn)
sensitivity <- tpr
specificity <- tn/(tn + fp)
kable(data.frame(sensitivity,specificity))
```

Table 6: Computed sensitivity and specificity for threshold = 0.6

sensitivity	specificity
0.7321429	0.8055556

4.3 Model Accuracy

```
(tp+tn)/dim(foo[1])[1]
```

```
[1] 0.785
```

Our model's accuracy is 78.5% for the chosen threshold.

4.4 Misclassification Rate

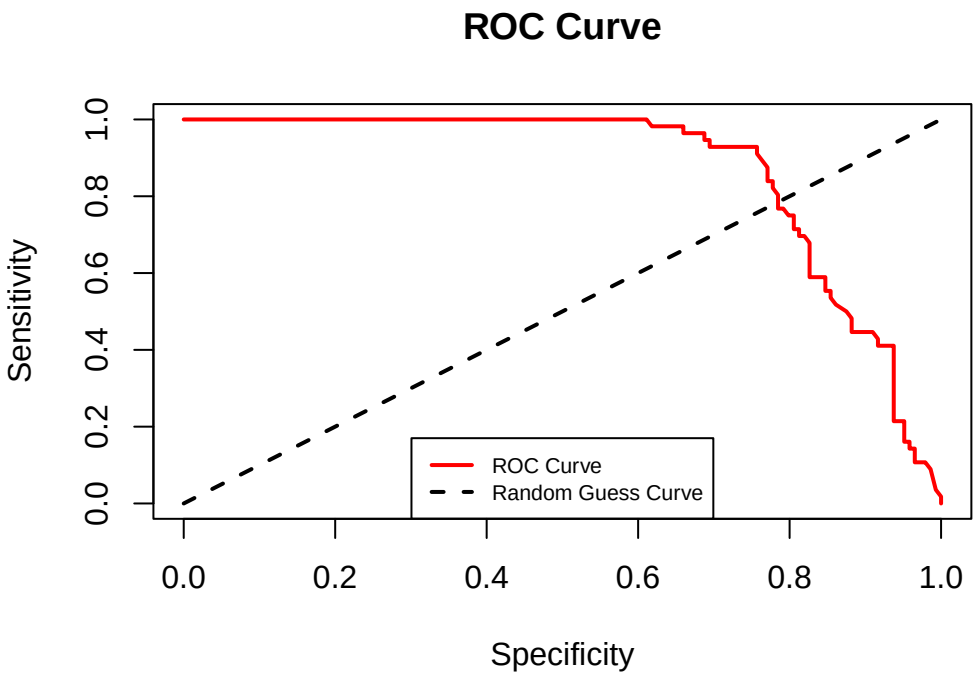
```
1 - (tp+tn)/dim(foo[1])[1]
```

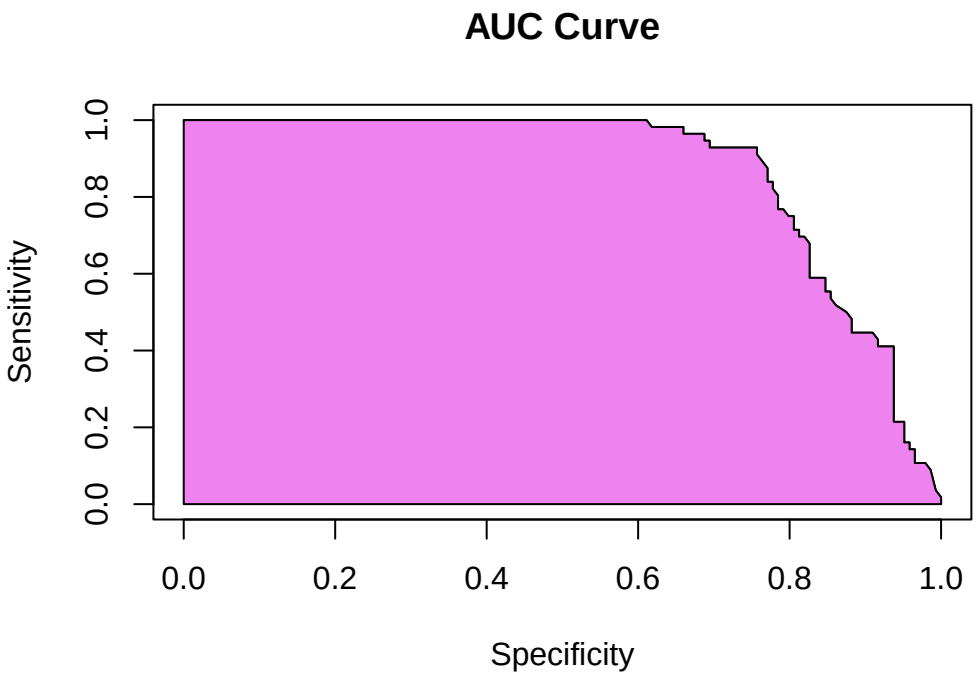
```
[1] 0.215
```

Our model's misclassification rate is 21.5% for the chosen threshold.

4.5 ROC and AUC Curves

```
threshold.vec <- seq(0, 1, length = 1e3)
sensitivity.vec <- numeric(length = 1e3)
specificity.vec <- numeric(length = 1e3)
for (i in 1:length(threshold.vec))
{
 threshold <- threshold.vec[i]
predict.y <- ifelse(pi.hat > threshold, 1, 0)
foo <- data.frame(y, predict.y)
tp <- sum(foo[ ,1] == 1 & foo[ ,2] == 1)
fn <- sum(foo[ ,1] == 1 & foo[ ,2] == 0)
fp <- sum(foo[ ,1] == 0 & foo[ ,2] == 1)
tn <- sum(foo[ ,1 ] == 0 & foo[, 2] == 0)
sensitivity.vec[i] <- tp/(tp + fn)
specificity.vec[i] <- 1 - (fp/(fp + tn))
}
dat <- data.frame(threshold.vec, specificity.vec, sensitivity.vec)
plot( dat[c(2,3)], type="l", col = "red", xlab = "Specificity",
      ylab = "Sensitivity", main = "ROC Curve", xlim = c(0,1), ylim = c(0,1), lwd = 2)
lines(x = specificity.vec, y = specificity.vec, lty = 2, col = "black", lwd = 2)
legend ("bottom", legend = c("ROC Curve","Random Guess Curve"),
        col = c("red","black"), lty = c(1,2), lwd = c(2,2), cex=0.7)
plot( dat[c(2,3)], type ="l", xlab = "Specificity",
      ylab = "Sensitivity", main = "AUC Curve")
polygon(x = c(min(dat[c(2,3)]$specificity.vec), dat[c(2,3)]$specificity.vec,
              max(dat[c(2,3)]$specificity.vec)),
        y = c(0, dat[c(2,3)]$sensitivity.vec, 0), col = "violet")
```


ROC Curve
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
ROC Curve
Random Guess Curve
Specificity


AUC Curve
Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
Specificity

4.6 Optimal Threshold

```
index <- dat$sensitivity.vec > 0.92 & (dat$specificity.vec >0.73 & dat$specificity.vec < 0.8)
kable(dat[index, ], row.names = FALSE)
```

Table 7: Choosing optimal threshold

threshold.vec	specificity.vec	sensitivity.vec
0.5815816	0.7361111	0.9285714
0.5825826	0.7361111	0.9285714
0.5835836	0.7361111	0.9285714
0.5845846	0.7569444	0.9285714

Optimal threshold is 0.5845846

Finding the model accuracy:

```
tpr <- tp/(tp + fn)
fpr <- fp/(fp + tn)
sensitivity <- tpr
specificity <- tn/(tn + fp)
(tp+tn)/200
```

```
[1] 0.72
```

Model accuracy is 72%.